

CS 595: Spring 2023

List of Potential Projects

Kyle C. Hale

This is a list of potential projects. This course is meant to get you involved in doing *research*. Accordingly, none of these projects are meant to be easy. They are all rather open-ended, and are meant to get you on the path to doing something new and original. There is no guarantee that such projects will be a success, but that is the nature of research. I want you to be ambitious! If you have your own ideas for a project that is within reason, we can of course discuss that as well. In general, I envision that your project will be in the context of edge computing, serverless, cloud, distributed computing, etc.. Note that a good number of these projects will involve developing *cross-cutting* knowledge in systems. You will be exposed to language and runtime design, OSES, virtualization, networking, and wireless technology. The descriptions below are intentionally vague. If you are interested in a particular project, but want more concrete directions, we can discuss.

- Serverless: Modify PyTorch to automatically offload ML computations to cloud/edge
- Serverless: Come up with a faster method of state management and sharing for serverless functions
- Serverless: Help benchmark and optimize serverless Virtines (virtual subroutines)
- Serverless: Modify a JIT compilation framework (Javascript, Python) to use Virtines
- Serverless: Extend virtines to isolate memory using Intel's Memory Protection Keys (MPK)
- Serverless: Implement better compiler support for virtines
- Serverless: Design a hardware accelerator to speed up virtines (especially snapshotting)
- Edge: Vehicle-to-vehicle offload (e.g. using RISC-V boards)
- Edge: Hand off delay-tolerant sensor readings to vehicles.
- Edge: Build a collaborative content cache for mobile device streaming
- Edge: Implement a collaborative tasking runtime at the edge (for example using WASM)
- Edge: Implement (or extend) a vehicle-to-vehicle (V2V) computing platform using WasmEdge
- Edge: Implement a drone detection system using crowdsourced data and edge computing, or SDR devices

- Edge: Improve the DEAN blockchain protocol
- Edge: Improve DART, e.g. by optimizing data shuffling services
- Edge/Serverless: Implement an environmental audio streaming engine using serverless
- Edge/Serverless: Build an edge analytics engine for pet vocalizations using edge/serverless
- Edge/Serverless: Use edge computing (or serverless) to offload perceptual algorithms for robotics (e.g., point cloud compression)
- Disaggregation: study interference of HPC apps in disaggregated environment
- Disaggregation: help implement more advanced compiler support for TrackFM remote memory framework
- Disaggregation: help extend TrackFM to support other languages: C++, Rust, etc.
- Disaggregation: explore heap allocation pruning for TrackFM